Tutorial 4: Comparisons of Groups

Tom Miller and Jason Pienaar

Statistics with R: the t-test

To this point, we have used R to read and edit data sets and to graphically explore our data. But, our primary interest for this course is using R to help statistically evaluate data, so this tutorial will cover using R to compare groups of continuous data. The simplest and best known way to compare two groups is the Student's t-test and this tutorial will also include an introduction to using ANOVA, which is a much more complicated and powerful hammer with which to hit this kind of data.

To begin with, read in the sailfin mollies data set in "eg1.txt" (same data set used in Tutorial 3). This file has six columns (POP, IDNO, RAYNO, SL, FINAREA, and TAREA). Fish were collected from 3 different populations (POP 1,2, and 5), with four measurements taken from each fish: ray number (RAYNO), standard length (SL), fin area (FINAREA), and tail area (TAREA). The last three variables are clearly continuous and appropriate for applying the t-test.

Let us propose the hypothesis that the standard length of the mollies in population 2 is either larger or smaller than for population 5. Since we do not assume a priori that either population is larger than the other, we would want to use a two tailed t test (if, for example, we wanted to test if population 2 is larger than population 5, we would used a one-tailed test). Before performing the t-test however, we would want to check that the parametric assumptions for the t test are met. These include:

1. The samples are drawn from normally distributed populations.
2. The samples are from populations with equal variances.
3. Another consideration is the presence of outliers, which can have strong effects on the results of statistical tests (i.e over or under inflate the test statistic).

To see if the samples are normally distributed, we could look at the distributions of the two variables (as well as their variances for the "equal variance" assumption). Thus we could begin by plotting histograms of the two variables

```
>hist(SL[POP==2])
>hist(SL[POP==5])
```

Based on these plots, we can see that the data appear to be roughly normally distributed. The tail to the left might concern us a bit. We could actually see it a bit better if we had more bars in our histogram. We can do this using the "*break*" option for *hist*:

```
>hist(SL[POP==2],breaks=10)
>hist(SL[POP==5],breaks=10)
```

Hmm, this also looks like we have a lot of tail to the left (not to be rude!). At this point (or even before we plot the histograms) we may want to plot box plots of the two variables to check if there is any hope of meeting the assumptions of the t-test:

```
>boxplot(SL[POP==2], SL[POP==5])
```

By the way, there is a short-cut that you can use to see all boxplots for all the subclasses of a variable:

```
>boxplot(SL~POP)
```

Neat, eh? Just use the tilda (~). This works for a lot of the graphics, but not generally for other functions.

With boxplots, the first assumption can be verified by observing if the data are spread symmetrically around the median). When we look at the top and the bottom of the boxes as well as the two whiskers we can see that they are all roughly equidistant from the median line, although the lower whiskers from both populations are a bit longer. Remember that the t-test is relatively robust to the normality assumption, so we will accept that the first assumption is met. Assumption checking is often done exactly like this, by looking at distributions and convincing yourself that they meet the assumptions, therefore it is up to you to be ethical about these kinds of things.

What about the second assumption? To calculate the variances and standard deviations of the two samples we could use the built in R functions *var* and *sd*. E.g.

```
>var(SL[POP==2])
>var(SL[POP==5])
>sd(SL[POP==2])
>sd(SL[POP==5])
```

When we do this we can see that, while the variance for population 5 is larger, populations 2 and 5 are relatively similar, so our simple 2 sample t test is probably appropriate.

Another thing to consider is the sample size of each group. To check the sample sizes, we can use the *length* function to determine the length of the vectors in which the samples are stored. E.g.

```
>length(SL[POP==2])
>length(SL[POP==5])
```

So, after checking our assumptions, we can finally do the darn t-test. To perform the actual test we can use the built in *t.test* function:

```
t.test(SL[POP==2],SL[POP==5])
```

R produces several lines of output for this command. Most importantly, it gives us a t value, the degrees of freedom, and the p-value. R also provides the means of each group and some other information. We will assume that we had agreed that a p-value < 0.05 would be significant (Type I error level) before we conducted the test. So, in this case, our two groups do not appear to be significantly different.

Now, we might have looked at the box plots and recognized that POP 5 had longer fish than POP 2. We could have crafted a slightly different hypothesis: that POP 5 was greater than POP2. This allows a one-tailed t-test, which requires the use of the *alternative* argument in R, asking if the first group is either "less" or "greater" than the second group. In our case, this would be:

```
t.test(SL[POP==2],SL[POP==5],alternative="less")
```

In this case, our groups are still not significantly different (or, more correctly, now population 5 has not been shown to be significantly greater than population 2).

Analysis of Variance

Remember that the t-test allowed us to compare two groups: our explanatory variable is categorical with two levels, addressing with the following hypothesis:

$$H_0: \mu_1 = \mu_2 \qquad H_A: \mu_1 \neq \mu_2$$

Analysis of variance allows us to ask a similar question, but we can have multiple levels or groups:

$H_0: \mu_1 = \mu_2 = u_3 = etc.$ $\qquad$ $H_A: \mu_k \neq \mu_l$ $\qquad$ for some k and l, where k is not equal to l

This last hypothesis is a little weird. But, look at the null hypothesis – we are asking if all the means are equal. Therefore, the null hypothesis is rejected if any of the means are different from any other, which gives us the odd notation for our H_A.

I am going to show you two ways to do a simple one-way ANOVA (that is, there is just one explanatory variable, although it may have many levels -- factorial designs, with >1 explanatory variable, will be dealt with later). First, we will use R to essentially do the analysis by hand, calculating all the necessary terms from our raw data. Then, we will use a much simpler command to do the same thing, much more quickly. There are two reasons for showing you the first somewhat tedious way to do a one-way ANOVA. We do want you to understand the variance terms that go into getting the F-values in

ANOVA. But, as important, we want you to think about the assumptions behind the tests before you blindly apply the simpler method.

A last note before we begin. The one-way ANOVA with two-levels is exactly equivalent to doing a simple t-test between two groups.

The data

We are going to use the pitcher plant data collected by this class in 2004. This is the same data you used for Exercise 1 and is on the webpage as "pitcherplant.txt". You should know how to read the file into R and make the columns accessible (remember to use *attach*). The results of this project were published in Ecological Entomology in 2007 (Hoekman, et al.).

Take a look at the file to remind yourself of the experiment. The class implemented three treatments (column "trt") of providing 0, 2, or 20 ants to each pitcher leaf. They then measured the response of different groups, including the number of mosquitoes and midges. For our ANOVA, we are interested in the response in mosquito oviposition rate, which is found in the "mosq.no" column.

This simple ANOVA is again a parametric test, so we are assuming that our data are normally distributed and with reasonably similar variances. It is always a good idea to take a look at the distributions using box plots and histograms. Let's make a nice little picture with all the appropriate graphs:

```
split.screen(c(2,2))
screen(1)
boxplot (mosq.no~ants)
screen(2)
hist (mosq.no[ants==0])
screen(3)
hist (mosq.no[ants==2])
screen(4)
hist (mosq.no[ants==20])
```

This should all be second nature to you by now. How do the distributions look? You should be able to see that the data appear skewed right; that is, the distributions have long tails toward high mosquito numbers. Try log-transforming the numbers – wait, we have a problem with zeros. Both the right-skewed data and the zero's are common problems in ecological data. A common solution is to add some constant value (often 1.0) to the data, then log-transform. Let's see what that looks like:

```
split.screen(c(2,2))
screen(1)
boxplot(log(mosq.no+1)~ants)
screen(2)
hist(log(mosq.no+1)[ants==0])
```

```
screen(3)
hist(log(mosq.no+1)[ants==2])
screen(4)
hist(log(mosq.no+1)[ants==20])
```

Hey, that looks a lot better, except for the 20 ants treatment group. Since ANOVA is fairly robust to the assumption of normality, we are going to plow forward.

ANOVA –the long-way

First, we need the "grand mean", the average mosq.no across all treatments (abbreviated as $\bar{y}$ in Quinn and Keough and the equations below). That is easily determined, but remember that we have to do the analysis on the transformed data:

```
gm=mean(log(mosq.no+1))
```

Then we need the mean and variances for each treatment level: that is, the mean mosq.no for the 0, 2, and 20 ant treatments. For convenience, I am going to put these in vectors:

```
grpMeans = c(mean(log(mosq.no[ants==0]+1)),mean(log(mosq.no[ants==2]+1)),
             mean(log(mosq.no[ants==20]+1)))
grpVars  = c(var(log(mosq.no[ants==0]+1)),var(log(mosq.no[ants==2]+1)),
             var(log(mosq.no[ants==20]+1)))
```

The total variation in mosquito number can be characterized by the total sums of squares. We get this by taking each data value (y_{ij} is the jth replicate observation from the ith group) minus the grand mean, squaring this value, then summing this up for all leaves (n) and all treatments (p):

$$SS_{total} = \sum_{i=1}^{p}\sum_{j=1}^{n}(y_{ij} - \bar{y})^2$$

We can use R as a big calculator as:

```
sst = sum((log(mosq.no+1)-gm)^2)
```

Go ahead and look at the SST. You should have found a value of 77.192. ANOVA procedure now partitions this total SS into residual and treatment components. The residual SS is the sum of the squares of the differences between the data points, y and their individual treatment means ($\bar{y}_i$):

$$SS_{error} = \sum_{i=1}^{p}\sum_{j=1}^{n}(y_{ij} - \bar{y}_i)^2$$

We can get R to determine this as:

```
ssw = sum((log(mosq.no[ants==0]+1)-grpMeans[1])^2) +
      sum((log(mosq.no[ants==2]+1)-grpMeans[2])^2) +
      sum((log(mosq.no[ants==20]+1)-grpMeans[3])^2)
```

The treatment SS is the sum of the squares of the differences between the individual treatment means and the grand mean:

$$SS_{treatment} = \sum_{i=1}^{p}\sum_{j=1}^{n}(\bar{y}_i - \bar{y})^2$$

which R can determine as:

```
ssb = (length(mosq.no[ants==0])*(mean(log(mosq.no[ants==0]+1))-gm)^2) +
      (length(mosq.no[ants==2])*(mean(log(mosq.no[ants==2]+1))-gm)^2) +
      (length(mosq.no[ants==20])*(mean(log(mosq.no[ants==20]+1))-gm)^2)
```

So the total SS should now be divided into two additive parts, the treatment SS and the residual SS. Here is where you can check if you have made some error. Does sst = ssb + ssw?? If not, you have made an error. Now, we compute the F statistic as:

$$Fstat = \frac{\frac{ssb}{(\#\ trts - 1)}}{\frac{ssw}{\sum_{i=1}^{k}(\#\ reps - 1)}}$$

Or, in R:

```
Fstat = (ssb/(3-1)) / (ssw/(23+24+24))
```

And, we can figure out the significance by calculating a p value from the F distribution with appropriate degrees of freedom, based on the number of treatments and replication within treatments. We do that using the *pf* function:

```
1 - pf (Fstat,df1=2,df2=71)
```

If you are still OK to this point, then you should be able to see that the null hypothesis of equal means is rejected. In other words, some of the treatments produced significantly different numbers of mosquitoes than others.

ANOVA – the short way

Remember the caveat that you need to really understand what ANOVA does to use short-cuts. But, this short-cuts is really simple, using the *aov* function.

```
fit=aov(log(mosq.no+1)~factor(ants),data=pp)
```

To see the results, just read the fit variable you have created, and its summary:

```
fit
summary(fit)
```

You should get:

```
> fit
Call:
   aov(formula = log(mosq.no + 1) ~ factor(ants), data = pp)

Terms:
                       factor(ants)    Residuals
Sum of Squares            10.72988     66.46245
Deg. of Freedom                2           71

Residual standard error: 0.967518
Estimated effects may be unbalanced

> summary(fit)
                Df      Sum Sq     Mean Sq     F value       Pr(>F)
factor(ants)     2      10.730       5.365      5.7312    0.004927 **
Residuals       71      66.462       0.936
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Check to make sure these values are the same as those you determined the long way.